ศรัณย์ ศรีศุภร

SOFT MAIL

การสื่อสารโดยใช้บริการของเน็ตแวร์แบบ IPX

อธิบายถึง
วิธีการสื่อสาร
โดยใช้ IPX
เพราะเป็นวิธีที่
มีประสิทธิภาพมากที่สุด

ในปัจจุบันมีการใช้งานระบบ LAN (Local Area Network) กันมาก ส่วนใหญ่จะใช้ซอฟต์แวร์ระบบแลนของบริษัทโนเวลล์ ซึ่งมีเน็ตแวร์เป็นตัวจัดระบบงานบนเน็ตเวอร์ก แต่เนื่องจากโปรแกรมที่ใช้งานโดยทั่วไปจะถูกใช้ไปในลักษณะการใช้ฐานข้อมูลร่วมกัน (Share Data) ซึ่งพัฒนามาจากโปรแกรมสำเร็จรูปบางตัว เช่น ฟอกซ์โปร/แลน หรือคลิปเปอร์ แต่การเขียนโปรแกรมในลักษณะการรับหรือส่งข้อมูลระหว่างเวอร์กสเตชันในระบบแลนยังเป็นสิ่งที่ยาก เนื่องจากการขาดแคลนหนังสือคู่มือประกอบ หรือการขาด ผู้เชี่ยวชาญเฉพาะด้านเกี่ยวกับเรื่องนี้ ที่จะนำความรู้ต่าง ๆ มาเผยแพร่

ในบทความต่อไปนี้ จะเป็นการแนะนำวิธีการเขียนโปรแกรมรับหรือส่งข้อมูล ระหว่างเวอร์กสเตชัน 2 เวอร์กสเตชันในระบบแลนวงเดียวกันของบริษัทโนเวลล์ด้วยโปรแกรมเน็ตแวร์ เวอร์ชัน 3.10 หรือมากกว่า การให้บริการของเน็ตแวร์ในการสื่อสารมีหลายวิธี ดังนี้

- ใช้ IPX
- ใช้ SPX
- ใช้ NetBios

ในที่นี้จะอธิบายวิธีการสื่อสารโดยใช้ IPX เนื่องจากเป็นวิธีที่มีประสิทธิภาพมากที่สุดในทั้ง 3 วิธี

Request and Reply Buffers

บางฟังก์ชันของเน็ตแวร์ต้องการแอดเดรสของบัฟเฟอร์ 2 ตัว บัฟเฟอร์ตัวแรกจะเรียกว่า request buffer จะ

ประกอบด้วยข้อมูลต่าง ๆ ที่จะถูกส่งไปในเน็ตเวอร์ก ส่วนบัฟเฟอร์ตัวที่สองจะเรียกว่า reply buffer ซึ่งจะใช้เก็บข้อมูลผลลัพธ์ หรือคำตอบของเน็ตเวอร์ก

Request Buffer

แอดเดรสของ request buffer จะถูกส่งผ่านโดยใช้รีจิสเตอร์คู่ DS:SI โดยปกติเวิร์ด (2 ไบต์) แรกของบัฟเฟอร์จะเก็บค่าความยาว (ขนาดหน่วยไบต์) ของบัฟเฟอร์

Reply Buffer

แอดเดรสของ reply buffer จะถูกส่งผ่านโดยใช้รีจิสเตอร์คู่ ES:DI โดยปกติเวิร์ด (2 ไบต์) แรกของบัฟเฟอร์จะเก็บค่าความยาว (ขนาดหน่วยไบต์) ของบัฟเฟอร์

IPX (Internetwork Packet Exchange)

เป็นรูปแบบการรับหรือส่งข้อมูลที่เป็นแบบ datagram ในการส่งข้อมูลจะต้องมีการบอกแอดเดรสปลายทางที่จะส่งด้วย และในการส่งข้อมูลออกไปจะไม่มีการรับรองความถูกต้องของข้อมูลที่จะส่งไปแล้ว

ทุกครั้งที่มีการรับหรือส่งข้อมูลจะถูกควบคุมโดย ECB (Event Control Block)

ตัวอย่าง

ก่อนที่โปรแกรม (application) จะสามารถส่งข้อมูล (packet) ได้ โปรแกรมจะต้องมีการเตรียม ECB ก่อน ต่อไปโปรแกรมจึงนำแอดเดรสของ ECB ไปไว้ในคู่เซกเมนต์รีจิสเตอร์ที่เหมาะสม แล้วจึงเรียก IPX เพื่อส่งข้อมูล

ESR (Event Service Routine) เป็นฟังก์ชันที่ถูกกำหนดขึ้นในโปรแกรม โดยฟังก์ชันนี้จะถูกเรียกใช้โดย ECB หลังจากที่มีเหตุการณ์ (event) เกิดขึ้น

Offset	Content	Type
0	Checksum	Word
2	Length	word
4	Transport Control	byte
*s 5	Packet Type	byte
*s 6	Destnation Network	byte[4]
*s 10	Destnation Node	byte[6]
*s 16	Destnation Socket	word
18	Source Network	byte[4]
22	Source Node	byte[6]
28	Source Socket	word

รูปที่ 1
โครงสร้าง IPX แพคเก็ตเฮดเดอร์

Type	ความหมาย
0	Unknow Packet Type
1	Routing Information Packet
2	Echo Packet
3	Error packet
4	Packet Exchange Packet
5	Sequenced Packet Protocol Packet

รูปที่ 2
ชนิดของ Pucket Type

โครงสร้าง (Structure) ของ IPX จะประกอบด้วยส่วนเฮดเดอร์ 30 ไบต์ และตามด้วย 0 ถึง 546 ไบต์ของข้อมูล (ในการส่งข้อมูลแบบ IPX แต่ละครั้งจะส่งได้มากที่สุด 546 ไบต์)

IPX Packet Header มีโครงสร้างดังรูปที่ 1

โปรแกรมจะต้องให้ค่าตัวแปร ในช่องที่มีเครื่องหมาย *s เมื่อจะส่งข้อมูล ส่วนตัวแปรในช่องที่เหลือจะถูกจัดการโดยตัว IPX เอง

- Checksum จะเก็บค่า checksum ของแพคเก็ตเฮดเดอร์ 30 ไบต์
- Length เก็บความยาวของข้อมูลทั้งหมด (มีค่า 0 ถึง 567 ไบต์)
- Transport Control มีค่าเป็น 0 ก่อนที่จะส่งข้อมูล
- Packet Type จะบอกชนิดของการให้บริการที่จะให้หรือรับข้อมูล ชนิดของการให้บริการ มีดังรูปที่ 2

เมื่อใช้การบริการ IPX ค่า packet type ควรจะมีค่าเป็น 4

- Destination Network และ Source Network เก็บค่าเน็ตเวอร์กแอดเดรสของปลายทาง และต้นทางตามลำดับ
- Destination Node และ Source Node เก็บค่าแลนบอร์ดของปลายทางและต้นทางตามลำดับ
- Destination Socket และ Source Socket เก็บค่าซ็อกเก็ตแอดเดรสของปลายทางและต้นทาง ตามลำดับ

โครงสร้างของ BcB มีรูปแบบดังนี้

ต้นฉบับไม่มีหน้า 257-258

No This Page 257 to 258 In Original

โปรแกรมจะต้องให้ค่าตัวแปร ในช่องที่มีเครื่องหมาย s เมื่อจะส่งข้อมูล และในช่องที่มีเครื่องหมาย r เมื่อจะรับข้อมูล

- ESR เก็บแอดเดรสของฟังก์ชัน ESR ที่ถูกกำหนดในโปรแกรม
- In Use เก็บค่าสถานะปัจจุบัน
- Completion Code จะเก็บค่า complete code ซึ่งใช้บอกว่าการรับข้อมูลเสร็จเรียบร้อย
- Immediate Address เก็บค่าแอดเดรสของโหนดโดยค่านี้จะได้จากฟังก์ชัน GetLocal Target
- Flagment Count เก็บจำนวนของบัฟเฟอร์ในแพคเก็ตค่าของ flagment count จะขึ้นอยู่กับจำนวนแฟล็กเมนต์แอดเดรสและแฟล็กเมนต์ไซด์ แต่ละค่าแฟล็กเมนต์แอดเดรสจะเก็บแอดเดรสของแต่ละบัฟเฟอร์ และค่าแฟล็กเมนต์ไซด์จะเก็บขนาดของแต่ละบัฟเฟอร์

การเรียกใช้บริการ IPX จะเรียกผ่านไปยังแอดเดรสที่ได้จากการเรียกใช้อินเตอร์รัพต์ 2Fh ฟังก์ชัน 7Ah ค่าของแอดเดรสนั้นจะดูที่คู่รีจิสเตอร์ ES:DI

ฟังก์ชันต่าง ๆ ที่ IPX มีบริการที่จำเป็นต้องใช้ มีดังต่อไปนี้

IPX Close Socket ปิด IPX ซ็อกเก็ต

ทางเข้า

BX = 01

DX = Socket Number

IPX Get Internetwork Address จะส่งค่าเน็ตเวอร์กแอดเดรสและโหนด-แฮดเดรสของเวอร์กสเตชันที่เรียกใช้

ทางเข้า

BX = 09

ค่าส่งกลับ

ES:SI = Reply buffer address

Reply buffer (10 byte)

Offset	Content	Type
0	Link Address	byte[4]
r,s 4	ESR Address	byte[4]
8	In Use Flag	byte
r,s 9	Completion Code	byte
10	Socket Number	word
12	IPX Workspace	byte[4]
s 16	Driver Workspace	byte[12]
s,r 28	Immediate Address	byte[6]
s,r 34	Flagment Count	word
s,r 36	Flagmeny Address 1	byte[4]
s,r 40	Flagment Size 1	word
s,r 42	Flagment Address 2	byte[4]
s,r 46	Flagment Size 2	word
	.....	

รูปที่ 3
โครงสร้างของ ECB

Offset	Content	Type
0	Network address	byte[4]
4	Node address	byte[6]

รูปที่ 4

IPX Get Local Target ฟังก์ชันนี้จะส่งค่า 6 ไบต์ โหนดแอดเดรสของ local traget

ทางเข้า

BX = 02

ES:SI = Request buffer address

DS:DI = Reply buffer address

ค่าส่งกลับ

AL = Completion Code

CX = Transport Time

ค่า Completion Code

00h = Successful

FAh = Not Path to Destination Node Found

IPX Listen for Packet รอรับข้อมูลจากฝ่ายส่ง

ทางเข้า

BX = 04

ES:SI = ECB address

ค่าส่งกลับ

AL = Completion Code

ค่า Completion Code

FFh = Listening Socket Dest Not Exist

IPX Open Socket เปิด IPX Socket

ทางเข้า

BX = 00

AL = 00

DX = Request Socket Number

ค่าส่งกลับ

AL = Completion Code

DX = Assign Socket Number

ค่า Completion Code

00h = Successful

FEh = Socket Table is Full

FFh = Socket Already open

IPX Send Packet ส่งข้อมูลทางเข้า

BX:SI = 03

ES:SI = ECB address

Offset	Content	Type
0	Destination Network	byte[4]
4	Destination Node	byte[6]
10	Destination Socket	word

รูปที่ 5

Offset	Content	Type
0	Local Target	byte[6]

รูปที่ 6

โปรแกรมตัวอย่าง PHONE

ในระหว่างที่เรากำลัง login อยู่ในระบบแลน เราอาจต้องการจะคุยกับเพื่อนร่วมงานบางคนที่ทำงานอยู่คนละชั้น เราต้องแน่ใจว่า เพื่อนคนที่เราต้องการคุยด้วยนั้นก็ login อยู่ในแลนเหมือนกัน เราก็ใช้โปรแกรม PHONE เพื่อคุยกับเพื่อนของเราได้ ในการคุยกันจะเป็นในลักษณะโต้ตอบกันทันทีทันใด ทั้งสองฝ่ายสามารถคีย์ข้อมูลพร้อมกันได้ ข้อความที่เรากดคีย์จะปรากฏให้เห็นทั้งสองฝ่าย

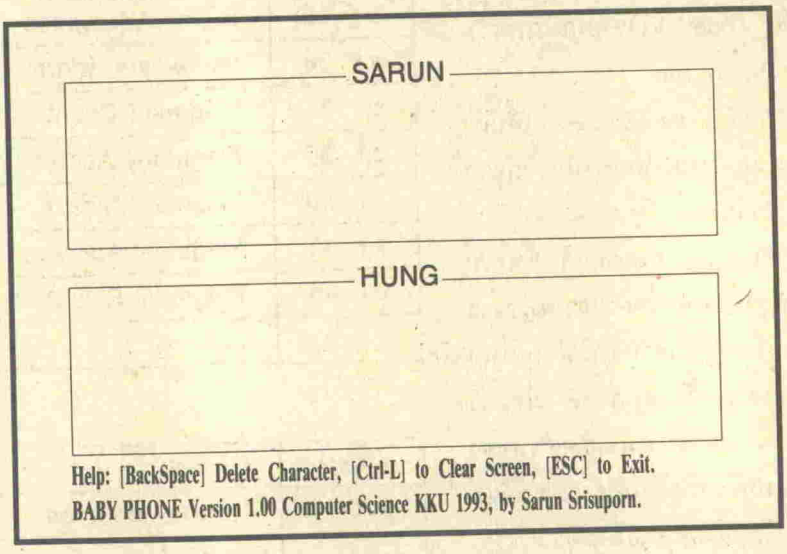
การทำงานของโปรแกรม PHONE จะมีอยู่ 2 ส่วน ฝ่ายแรกเรียกว่า ฝ่ายถาม (เป็นคนโทร PHONE) ฝ่ายหลังเป็นฝ่ายตอบรับ (เป็นคนตอบรับ ANSWER)

ฝ่ายถาม จะเรียกโปรแกรม PHONE.EXE และตามด้วยชื่อ username ของคนที่เราต้องการคุย

ฝ่ายตอบ จะเรียกโปรแกรม ANSWER.EXE ฝ่ายตอบจะรู้ว่ามีคนถามสังเกตโดยจะมีข้อความ

"**>> PHONE SARUN waiting to answer from you, type "ANSWER".**"

ปรากฏที่บรรทัดล่างสุดของหน้าจอ (เหมือนถูกใช้คำสั่ง SEND) ถ้าฝ่ายตอบ ต้องการคุยด้วย จะเรียกโปรแกรม ANSWER.EXE ที่ DOS Prompt หลังจากนั้นจะปรากฏหน้าจอขึ้นมาแสดงผลและรับข้อมูลข้อความที่คุยโต้ตอบกัน การยกเลิกการคุยกัน โดยกดคีย์ ESC



รูปที่ 7

ตัวอย่าง

การใช้โปรแกรม PHONE สมมุติฝ่ายถามชื่อ SARUN และฝ่ายตอบชื่อ HUNG

ที่ฝ่ายถาม เมื่อต้องการจะคุยกับ HUNG เรียกโปรแกรม

F:\SARUN>PHONE HUNG กด Enter จะปรากฏข้อความ

"Waiting to answer from SARUN..., [Esc] Exit."

ที่ฝ่ายตอบ เมื่อ SARUN phone มาแล้ว ที่บรรทัดล่างสุดของหน้าจอ จะปรากฏข้อความ

">> PHONE SARUN waiting to answer from you, type "ANSWER"."

ฝ่ายตอบจะกดคีย์ Ctrl-Enter เพื่อลบข้อความนั้นออกก่อน แล้วจึงเรียกโปรแกรม

F:\HUNG>ANSWER กด Enter

จะปรากฏข้อความ

"Waiting to connect from ..., [Esc] Exit."

จากนั้นหน้าจอทั้งสองฝ่ายจะปรากฏดังรูปที่ 7.เพื่อรับข้อมูล และแสดงผลข้อมูล เมื่อฝ่ายใดฝ่ายหนึ่งกดคีย์ ESC ก็จะยกเลิกโปรแกรมนี้ทันทีทั้งสองฝ่าย

หมายเหตุ

ซอร์สโปรแกรมและโปรแกรมใช้งานจริง ทางวารสารได้จัดเตรียมไว้ในซอฟต์เมล์ไมโคร 94